IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier XXXX

ChatGPT vs. Modest Large Language Models: an extensive study on benefits and drawbacks for conversational search

GUIDO ROCCHIETTI[1,2], Cosimo Rulli[1], Franco Maria Nardini[1], Cristina Ioana Muntean[1], Raffaele Perego[1] and Ophir Frieder[3]

[1]ISTI-CNR, Via G. Moruzzi 1, 56124 Pisa, Italy
[2]Department of Computer Science, University of Pisa, Largo B. Pontecorvo, 3 56127 Pisa, Italy
[3]Georgetown University, 3700 O St NW, Washington, DC 20057, United States

Corresponding author: Guido Rocchietti(e-mail: guido.rocchietti@isti.cnr.it).

Funding for this research has been provided by CAMEO, MIUR-PRIN 2022 n. 2022ZLL7MW, PNRR - M4C2 - Investimento 1.3, Partenariato Esteso PE00000013 - "FAIR - Future Artificial Intelligence Research" - Spoke 1 "Human-centered AI" funded by the European Union (EU) under the NextGeneration EU programme, and by the Horizon Europe RIA "Extreme Food Risk Analytics" (EFRA) funded by the European Commission under the NextGeneration EU programme grant agreement n. 101093026. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the EU or European Commission-EU. Neither the EU nor the granting authority can be held responsible for them.

ABSTRACT Large Language Models (LLMs) are effective in modeling text syntactic and semantic content, making them a strong choice to perform conversational query rewriting. While previous approaches proposed NLP-based custom models, requiring significant engineering effort, our approach is straightforward and conceptually simpler. Not only do we improve effectiveness over the current state-of-the-art, but we also curate the cost and efficiency aspects. We explore the use of pre-trained LLMs fine-tuned to generate quality user query rewrites, aiming to reduce computational costs while maintaining or improving retrieval effectiveness. As a first contribution, we study various prompting approaches — including zero, one, and few-shot methods — with ChatGPT (e.g., `gpt-3.5-turbo`). We observe an increase in the quality of rewrites leading to improved retrieval. We then fine-tuned smaller open LLMs on the query rewriting task. Our results demonstrate that our fine-tuned models, including the smallest with 780 million parameters, achieve better performance during the retrieval phase than `gpt-3.5-turbo`. To fine-tune the selected models, we used the QReCC dataset, which is specifically designed for query rewriting tasks. For evaluation, we used the TREC CAsT datasets to assess the retrieval effectiveness of the rewrites of both `gpt-3.5-turbo` and our fine-tuned models. Our findings show that fine-tuning LLMs on conversational query rewriting datasets can be more effective than relying on generic instruction-tuned models or traditional query reformulation techniques.

INDEX TERMS Conversational Search, Query Rewriting, Large Language Models, Instruction-tuned LLMs, Fine-tuning

STATEMENTS AND DECLARATIONS

This manuscript is an extension of our Web Intelligence and Intelligent Agent Technology 2023 contribution entitled "Rewriting Conversational Utterances with Instructed Large Language Models" [14]. In this extension, we build on top of it by investigating the impact of using open-source fine-tuned LLMs in a Conversational Query Rewriting task. Moreover, we propose a solution that shows improved results with respect to the previous work while considerably reducing its computational impact.

I. INTRODUCTION

Conversational Query Rewriting is a main task within the realm of Conversational Search. Conversational Search is a machine-human interface paradigm where the user interacts with a chatbot through a multi-turn, dialogue-like conversation. Rather than submitting queries by keywords — as in standard search engines — the seek for information happens by posing questions using natural language. In this framework, the user asks multiple questions relating to the contextual information expressed in previous interactions. When answering a question, the conversational system needs

to take the context of the entire conversation into account and output a well-formed answer using natural language. In particular, the peculiarities of conversational utterances, including the absence of context from previous questions, topic changes [31], [32], and inferred concepts from previous responses, affect the efficacy of traditional information retrieval methods. The purpose of query re-writing is to address linguistic phenomena such as ellipsis and omissions, while maintaining useful information expressed in previous queries [64]. Therefore, a rewriting system must grasp all dependencies throughout the user-machine interaction and reformulate the user query, generating a self-explanatory query that contains all the information necessary to retrieve the most relevant documents for the user request. These requirements make Large Language Models (LLMs) a perfect candidate for query re-writing.

The effectiveness of LLMs in many natural language processing (NLP) tasks is widely acknowledged; they are capable of delivering unprecedented performance in tasks such as summarization [19], question answering [18], [24], machine translation [53], [67], and sentiment analysis [68], [69]. The most famous example is ChatGPT [47], a proprietary model from OpenAI released in 2023. ChatGPT has demonstrated significant effectiveness in a variety of NLP tasks, such as generating coherent and contextually appropriate text, understanding and responding to complex queries, and performing detailed text analysis. Its success in these areas can be attributed to its large-scale architecture, extensive training data, and advanced fine-tuning techniques, which enable it to understand and generate human-like text.

Our current research builds upon the findings presented in our previous work [14], which focuses on using Instruction-tuned LLMs (ILLMs) to rewrite conversational queries formulated by users. As a first step, we test ChatGPT on the task of query rewriting. We use the `gpt-3.5-turbo` version of ChatGPT.[1] We instruct it to rewrite user queries in a conversation by proposing a prompting approach. On the one hand, we formulate many prompts to study their effectiveness when producing rewrites; on the other hand, we try different prompts with zero-shot and few-shot approaches, evidencing an increase in rewriting accuracy when providing the Instruction-tuned LLM with examples of rewritings.

In this previous work, we address the following research questions:

RQ1: How do instruction-tuned LLMs perform in rewriting conversational queries?

RQ2: Which prompting approach better improves conversational search results?

In this manuscript, the novel and unpublished contribution lies in replacing ChatGPT with cheaper, in terms of resource demands, open-source, thus increasing transparency and reproducibility of the model, fine-tuned LLMs. In this way, we study whether smaller, open-source models can be employed in a task such as Conversational Query Rewriting, and, at the same time, study the correlation between model size and effectiveness.

Within the aforementioned framework, we address the following research questions:

RQ3: Can a small, open-source LLM fine-tuned on the specific rewriting task perform as well as an optimally prompted instruction-tuned LLM?

RQ4: Which model in the above scenario provides the best efficiency-effectiveness trade-off?

Our goal is to improve the rewritings while decreasing the computational costs by fine-tuning open-source pre-trained LLMs. Fine-tuning allows for an in-depth specialization of models for the specific task. Moreover, using open LLMs over closed, proprietary ones offers several advantages for users and stakeholders in general. First, open models provide greater transparency, allowing users to study how the models make decisions by providing access to the whole model, which can improve trust and help to comply with regulatory standards. Additionally, open-source LLMs are highly customizable, providing the flexibility to fine-tune them for specific tasks or domains, unlike proprietary models that may limit modifications. They also offer users more control over their AI infrastructure and reduce long-term costs. Additionally, the open-source community often provides faster innovation cycles, with continuous updates and improvements driven by a broad base of contributors. Finally, open LLMs offer more flexible deployment options, whether on-premise or in the cloud, allowing to optimize security, privacy, and performance.

For our experiments, we select a dataset that represents the task we wish to perform and use it to teach the models to do the same. We vary sizes and architectures for LLMs and fine-tune them to rewrite conversational queries, evaluating them on the conversational datasets provided by TREC [8]. We obtain quality rewrites, with results for custom metrics that exceed state-of-the-art similar works [34] [61] [52]. At the same time, we apply state-of-the-art techniques such as quantization [10], not only to fine-tune models but also to observe the decrease in performance vs. the gain in terms of computational resources.

Finally, we study how different models can reach different points in the efficiency-effectiveness space. We observe that the highest contribution is reached by bigger models but, in terms of average inference time vs. quality of rewriting, we observe that the optimal trade-off can be achieved using smaller models based on T5 [39]. Furthermore, we study the contribution of the models throughout the conversation, observing that the highest contribution of the models is given for the last queries of the conversation.

To summarize, our contributions are as follows.

- We study the capabilities of Instruction-tuned LLMs in rewriting conversational queries by designing different prompting strategies and exploiting the capability of retaining contextual information to rewrite user queries.

[1]From now on, when referring to ChatGPT, we always refer to the `gpt-3.5-turbo` model.

- After establishing the capabilities of ILLMs to rewrite queries, we fine-tune several state-of-the-art LLMs using a dataset (QReCC [4]) specifically created for query rewriting, among other tasks, showing the impact of the size of the selected models on both rewrite quality and inference time.
- Finally, we test all models, both the ILLM and the fine-tuned ones when used in a conversational retrieval pipeline. To assess the improvements, we use the TREC CAsT datasets, which were specifically devised to study the retrieval effectiveness of conversational search systems. We show that by using ILLMs we increased precision up to 31.7% with respect to the state-of-the-art (e.g., ConvGQR [34]). Moreover, we achieve even better metric values when evaluating the fine-tuned models. We show an increase up to 10.58% for NDCG@3 with respect to the ILLM results.

The paper is structured as follows. Section II introduces the methodology used throughout the research. Section III describes the experimental evaluation adopted for our experiments. Sections IV and V present the results obtained during the two experimental phases. Finally, Section VI provides an overview of the state of the art, while Section VIII concludes the work and outlines the future steps of the research.

II. METHODOLOGY

We now overview our methodology. First, we define the problem, namely Conversational Query Rewriting. We then evaluate the rewriting accuracy of Instruction-tuned LLMs, i.e., models trained to take prompts as input and produce rewritten conversational queries. We then select several specific models with different sizes and pretraining, fine-tune them, and evaluate gains. In Table 1, we report the notation used throughout the paper.

TABLE 1. Notation.

Symbol	Definition
$\mathcal{U}$	A multi-turn conversation composed of a sequence of utterances asked by a user to a conversational assistant.
Θ	An Instruction-tuned LLM we use for utterance rewriting also referred to as *Assistant*.
Φ	A pretrained or fine-tuned LLM that we use to rewrite utterances.
u_i	The current original utterance at turn i in $\mathcal{U}$.
$\hat{u}_i$	The current utterance rewritten by Θ or Φ.
$u_1, \ldots, u_{i-1}$	The previous original utterances in $\mathcal{U}$.
$\hat{u}_1, \ldots, \hat{u}_{i-1}$	The previous utterances in $\mathcal{U}$ rewritten by Θ or Φ.
$\bar{u}_1, \ldots, \bar{u}_{i-1}$	The previous manually-rewritten utterances in $\mathcal{U}$.
$\hat{r}_1, \ldots, \hat{r}_{i-1}$	Responses to the previous utterances generated by Θ.
$\mathcal{C}$	The *Context* which is composed of the alternation between $u_1, \ldots, u_{i-1}$ and $\hat{u}_1, \ldots, \hat{u}_{i-1}$, or even adding $\hat{r}_1, \ldots, \hat{r}_{i-1}$. An example can be seen in Figure 1.
$\mathcal{E}$	The *Example* comprises original utterances $u_1, \ldots, u_{i-1}$ and their corresponding manually rewritten utterances $\bar{u}_1, \ldots, \bar{u}_{i-1}$.
s	The scope that explains our goal to the rewriting LLM Θ, also referred to as *System*.
p	The actual Prompt that, given u_i, specifies the instruction to Θ, namely, to rewrite the query.

A. INSTRUCTION-TUNED LLMS

We assess the rewriting capabilities of an Instruction-tuned LLM, investigating the impact of different prompts and instructions on the effectiveness of a two-stage conversational search pipeline.

A typical rewriting request consists of the following:

$$\Theta(s, \mathcal{E}, \mathcal{C}, p, u_i) = \hat{u}_i, \tag{1}$$

where s represents the scope, i.e., the general task instructions of how we want the system to behave, $\mathcal{E}$ is a different conversation example from the current one, $\mathcal{C}$ is the context of u_i, and p is the prompt accompanying u_i, which explicitly instructs Θ detailing the request for rewriting by adding specific desired characteristics, for example, "concise", "verbose", and "self-explanatory".

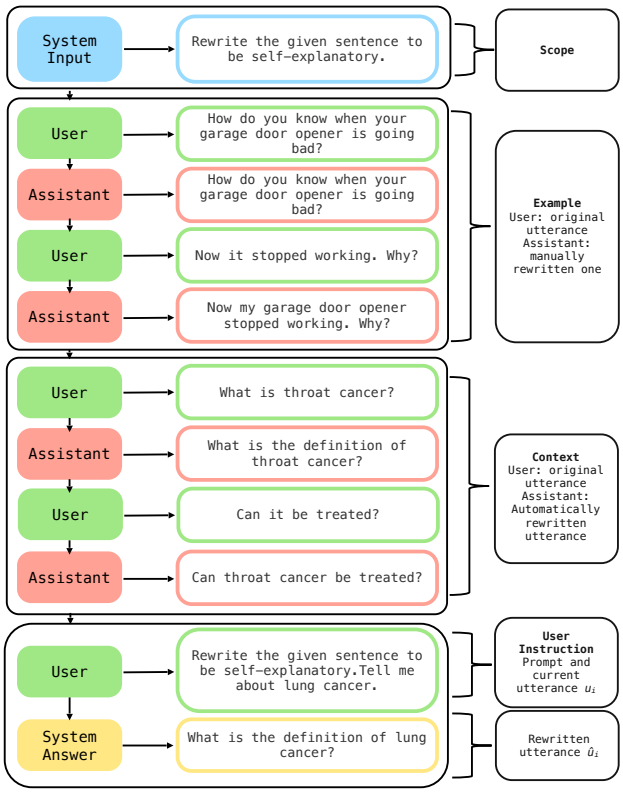


FIGURE 1. Main elements of an utterance rewriting request. The *Scope* indicates the task that the model should perform. The *Example* is the artificial part of the interaction where the user part is the query to rewrite, and the assistant part is the query rewritten by a human. The *Context* is composed of the previous queries rewritten by our model. The last section represents the current prompt and the output of the system.

In Figure 1, we present a visual example of a typical rewriting request. We can see how the first block represents the system, the second the example, and the third the context of the current conversation, while the last component contains the prompt and current question, followed by the answer (rewritten utterance) provided by the assistant.

Finally, in Figure 2, we show the experimental pipeline used to test the capabilities of Instruction-tuned LLMs to

rewrite conversational utterances. We provide the ILLM with different prompts and with the conversations extracted from the evaluation set and ask the model to rewrite them to perform an evaluation phase.

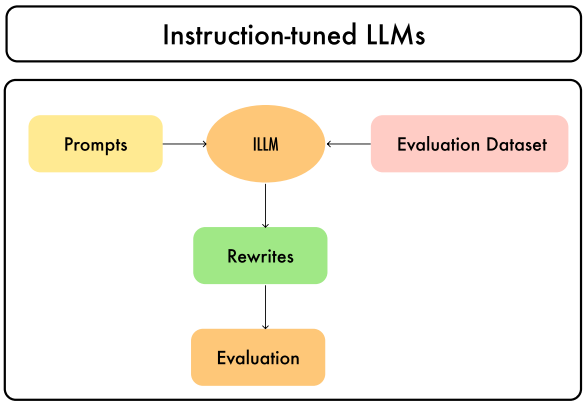


FIGURE 2. Experimental pipeline of the Instruction-tuned LLMs. The utterances are extracted from the evaluation dataset (TREC CAsT) and provided to the Instruction-tuned LLM with a prompt to produce rewrites. Finally, the evaluation phase consists of the retrieval and reranking phases.

B. PROMPTING CHATGPT

A typical request submitted through the ChatGPT API[2] contains the elements detailed in Eq. 1, namely, scope, example, context, prompt, and current utterance. For all prompts, the example $\mathcal{E}$ consists of an exemplary conversation, chosen randomly from one of the TREC CAsT datasets, not used for the evaluation, and not related to $\mathcal{U}$, where the user inputs are the original utterances, and the assistant inputs are instead the same utterances rewritten manually. Furthermore, the context $\mathcal{C}$ consists of the previous utterances of $\mathcal{U}$, where the user inputs are the original utterances $u_1, \ldots, u_{i-1}$, and the assistant inputs are instead the same utterances rewritten by the model, $\hat{u}_1, \ldots, \hat{u}_{i-1}$.

C. FINE-TUNING LLMS FOR QUERY REWRITING

After establishing the quality of the rewritings provided by `gpt-3.5-turbo`, we observe the rewriting capabilities of LLMs when fine-tuning them. A typical model for query rewriting consists of the following:

$$\Phi(C, u_i) = \hat{u}_i \tag{2}$$

Given only the context, which can also be empty in cases of first utterances in the conversation flow, and the current user query, we desire a model that can generate a rewritten and more complete version of the same request. To learn this model, we perform a fine-tuning phase.

Given a fine-tuning dataset $D = \{\mathcal{U}_1, \ldots, \mathcal{U}_n\}$, where each conversation $\mathcal{U}_j$ is composed of the original interactions between the user and the system $u_1, \ldots, u_{i-1}$, the current utterance that we wish to rewrite u_i, and the rewritten version of each utterance $\hat{u}_1, \ldots, \hat{u}_i$, we organize these data to create an input to fine-tune the model. To do so, we introduce three special tokens i_{token}, q_{token}, and a_{token} dilimiting the previous utterances $u_1, \ldots, u_{i-1}$, the current utterance u_i, and the rewrite $\hat{u}_i$, respectively.

We are now able to build the input x and the expected output y_t to fine-tune any type of LLM as follows:

$$x_i = i_{token} + u_1, \ldots, u_{i-1} + q_{token} + u_i + q_{token} \tag{3}$$

$$y_t = \hat{u}_i \tag{4}$$

In this way, after selecting a model to fine-tune Φ, we provide it with both $\mathbf{x} = \{x_1, \ldots, x_n\}$ and $\mathbf{y_t} = \{y_{t1}, \ldots, y_{tn}\}$, for it to learn how to rewrite conversational utterances by calculating the distance between the generated output y_i and the target one y_{ti}. The objective is the following:

$$\Phi(\mathbf{x}, \mathbf{y_t}) = \hat{\Phi}, \tag{5}$$

where $\hat{\Phi}$ is the model fine-tuned to rewrite conversational utterances, that is, $\hat{\Phi}(x_i) = \hat{u}_i$.

To do so, we use PyTorch and the Transformers Python libraries. We create models $\hat{\Phi}_{1,\ldots,k}$, specifically trained to rewrite conversational utterances without prompts. In Figure 3, we show the fine-tuning pipeline of our experiments. From the chosen dataset, we extract the context, the current utterance u_i, and the rewritten version $\hat{u}_i$ that we provide to the model to learn how to rewrite questions. Similarly, from the evaluation dataset, we extract the conversational utterances, rewrite them with the fine-tuned model, and evaluate them to assess the model's rewriting capabilities.

In practice, when using a tokenizer, we select three special tokens from the Falcon Tokenizer (i.e. »`INTRODUCTION`« as i_{token}, »`QUESTION`« as q_{token}, and »`ANSWER`« as a_{token}) and add them to the tokenizers of each selected model. At the same time, we organize the input data by selecting only the original previous questions $u_1, \ldots, u_{i-1}$ and using them as the context.

Motivating this choice is the ability to rely only on the conversational data provided by the user. This means that providing more input data (e.g., the answers) to the models may result in better rewrites, but our goal is to study the generative capabilities of the models when called in a low-information setting. This is also done to compare the results with those obtained using `gpt-3.5-turbo`, for which we did not provide further information such as the rewritten queries or even the answers.

Unlike the Instruction-tuned LLMs approach, here we define a method that does not rely on prompting to generate rewrites. This means avoiding the dependence of the different ways of formulating prompts and providing us with a more sound and less fluctuating way of performing rewritings. Similarly, relying on open-source models allows them to run on private machines and also to further modify and study the models to increase their capabilities. At the same time,

[2]https://platform.openai.com/docs/api-reference

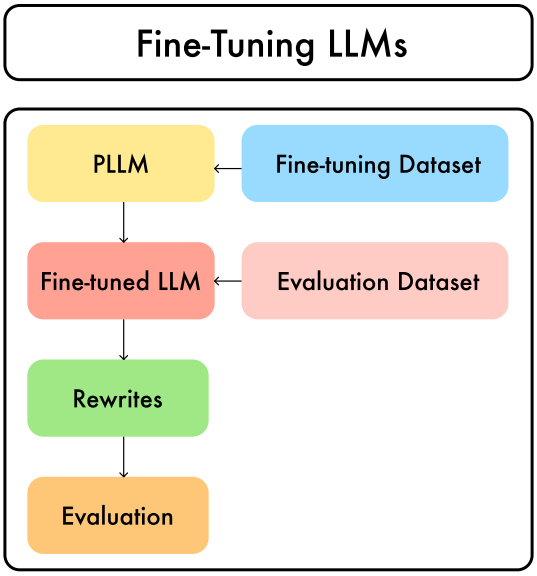


FIGURE 3. Experimental pipeline of the fine-tuning phase. The training data are extracted from the fine-tuning dataset (QReCC) and provided to the pre-trained LLM with a rewriting target to learn. After training the model, we produce the rewrites of the utterances from the evaluation dataset (TREC CAsT). Finally, we evaluate them by performing two steps: the retrieval and reranking phases.

as we will discuss thoroughly in Section V, we manage to significantly reduce the computational resources required to generate valuable utterance rewrites.

III. EXPERIMENTAL EVALUATION

In this section, we describe in detail the datasets used in the evaluation and the fine-tuning phases. We then describe the models selected for the fine-tuning part of the experiments and the post-processing phase. Finally, we describe the two-stage retrieval phase we employ to evaluate the quality of the rewrites.

A. CONVERSATIONAL DATASETS

Our experiments are based on two different sets of data sets. On the one hand, we use the TREC Conversational Assistant Track (CAsT) 2019 and 2020[3] datasets for the evaluation phase of both the Instruction-tuned LLM and the fine-tuned ones. On the other hand, we use the QReCC dataset for the fine-tuning phase of the models.

a: Evaluation Dataset

For the evaluation phase of both the Instruction-tuned LLMs and the Fine-tuned LLMs we use two datasets from the TREC Conversational Assistance Track, the 2019 and 2020 versions. The CAsT 2019 [9] dataset consists of 20 human-assessed test conversations, while CAsT 2020 [7] includes 25 conversations, with an average of 10 turns per conversation. The CAsT 2019 and 2020 datasets include relevance judgments at the passage level. Conversations are provided with original and manually-rewritten utterances. The manually-rewritten utterances are the same conversational utterances as the original ones, where human assessors resolve missing keywords or references to previous topics. Relevance judgments have a three-point graded scale and refer to passages of the TREC CAR (TREC Complex Answer Retrieval), the MS-MARCO (MAchine Reading COmprehension) and the WaPo (TREC Washington Post Corpus) collections for CAsT 2019 and 2020 for a total of 38,636,520 passage.

In these datasets, questions within a conversation are characterized by anaphora and ellipses. They imply a large part of the context and miss explicit references to the current topic. Table 2 reports some examples of utterances from the CAsT 2019 dataset. We can see that manually-rewritten utterances are concise and rephrase the original utterance by adding the missing tokens to make it self-explanatory. On the other hand, depending on the prompt, automatically-rewritten utterances tend to be more verbose although well-formed natural language questions.

b: Training Dataset

QReCC, or Question Rewriting in Conversational Context, is a dataset designed to answer open-domain questions. It is made up of 14,000 conversations, which include 81,000 question-answer pairs.

The dataset is derived from the questions found in TREC CAsT, QuAC, and Google Natural Questions. While the first two are multi-turn conversation datasets, the latter is not. QReCC aims to provide a benchmark for end-to-end conversational question answering. This includes question rewriting, passage retrieval, and reading comprehension. In the dataset, each query is rewritten by resolving references and other linguistic phenomena, resulting in a context-independent version of the original context-dependent question. Each of the queries is also annotated with an answer and a link to the web page that provided the answer. Each conversation in the dataset has a unique *Conversation_no*, *Turn_no* unique within a conversation, the original Question, Context, Rewrite, and Answer with *Answer_URL*.

It is important to note that the data coming from TREC CAsT was excluded from the QReCC data set to avoid the possibility of injecting biases inside the fine-tuned models. To address the equation 5 defined in Section II, we select QReCC as fine-tuning dataset D. We then create an ad hoc version of QRECC to carry out our experiment on different models, deploying a unique way of organizing the training data and the necessary tools. In Figure 4, we show how the fine-tuning pipeline works following the methodology explained in Subsection II-C.

For reproducibility reasons, we published the QReCC dataset organized as described on the HuggingFace page[4]. Similarly, all fine-tuned models are publicly available on HuggingFace [5].

[3]Conversational Assistant Track, https://www.treccast.ai/

[4]https://huggingface.co/datasets/giuid/qrecc_raw_context

[5]https://huggingface.co/giuid

TABLE 2. Example of conversation taken from the CAsT 2019 evaluation dataset. The "qid" column identifies the conversational turn, the first number refers to the conversation, and the second to the dialogical turn. The "Raw utterance" column reports the original utterance, the "Manually-rewritten utterance" column reports the human assessor rewritings, and the "Automatically-rewritten utterance" column reports an example of rewriting generated by ChatGPT with our proposed prompt P5.

qid	Raw utterance	Manually-rewritten utterance	Automatically-rewritten utterance
31_1	What is throat cancer?	What is throat cancer?	What is throat cancer?
31_2	Is it treatable?	Is throat cancer treatable?	Is throat cancer treatable?
31_3	Tell me about lung cancer.	Tell me about lung cancer.	Tell me about throat cancer.
31_4	What are its symptoms?	What are lung cancer's symptoms?	What are the symptoms of throat cancer?
31_5	Can it spread to the throat?	Can lung cancer spread to the throat?	Can throat cancer spread to other areas of the throat?
31_6	What causes throat cancer?	What causes throat cancer?	What causes throat cancer to spread to other areas of the throat?
31_7	What is the first sign of it?	What is the first sign of throat cancer?	What are the early signs of throat cancer?
31_8	Is it the same as esophageal cancer?	Is throat cancer the same as esophageal cancer?	Is throat cancer the same as esophageal cancer?
31_9	What's the difference in their symptoms?	What's the difference in throat cancer and esophageal cancer's symptoms?	What are the differences in the symptoms of esophageal cancer and throat cancer?

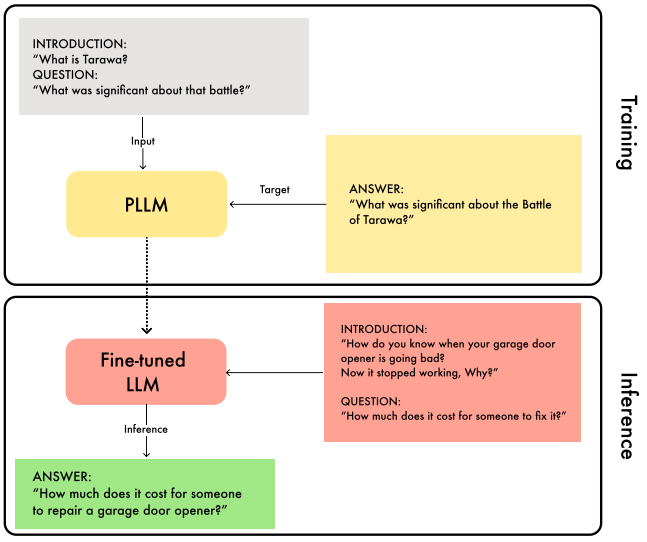


FIGURE 4. Fine-tuning pipeline. We provide the input, constituted by the previous utterances, the utterance to rewrite, and the target, i.e., the rewritten expected outcome. After the fine-tuning phase, we obtain a new fine-tuned model to which we provide the previous utterances and the current one from the evaluation dataset to produce the rewrites.

B. FINETUNING LLMS

To fine-tune several large language models (LLMs) on the QReCC dataset, we use PyTorch and the HuggingFace Transformers library. The QReCC dataset contains conversational questions and answers from various domains and is suitable for teaching LLMs to handle conversational context and topic shifts. We include the previous questions of the same conversation in the input to provide additional context for the LLM.

1) The models

The models we fine-tune are Mistral-7B, Falcon-7B, Llama-7B, 13B, Flan T5 large and XL. These models have different architectures, sizes, and pretraining objectives. The first model we consider, one of the few available when we started this research, is Falcon-7B. As said above, we select the special tokens that we use for all the models from the Falcon tokenizer. For all models, we use the Adam optimizer with a learning rate of 5e-06 and a batch size of 32. We run the fine-tuning for 10 epochs and evaluate the models on the QReCC validation set to find the best model. We now overview the selected models and their main characteristics.

All the models listed below are available in the HuggingFace repository.

- *Llama-2* is a collection of large language models (LLMs) that vary in size, with the largest having up to 70 billion parameters. These models are available both pre-trained and fine-tuned, with the fine-tuned versions optimized specifically for dialogue use cases. For our experiments, we tested the pretrained versions with different settings: 7 and 13 billion parameters.
- *Falcon-7B* is a decoder-only language model developed by the Technology Innovation Institute, with 7 billion parameters. Its architecture is based on the GPT-3 paper, but it includes unique features such as rotary positional embeddings, FlashAttention, and multiquery for efficient inference. Falcon-7B is available under the Apache 2.0 license, which allows unrestricted commercial use.
- *Mistral-7B* is a language model created by Mistral AI that features 7.3 billion parameters. It was trained on the Leonardo supercomputer and incorporates advanced techniques like Grouped-query attention and Sliding Window Attention for efficient inference. These methods enable the model to efficiently manage larger sequences and respond to multiple queries at once. In terms of performance, Mistral-7B is designed to surpass certain benchmarks set by Meta's Llama models.
- *FLAN-T5*, T5, short for Text-to-Text Transfer Transformer, is a model that uses a unique text-to-text approach. Instead of designing specific architectures for each task, T5 handles every task as a text generation problem. This includes tasks such as translation, question answering, and classification.
  The architecture of T5 is an encoder-decoder framework.

FLAN-T5 is an advanced version of the original T5 model, fine-tuned for a range of tasks. The FLAN-T5 variants include small, base, large, XL, and XXL, each differing in size. For our experiments, we tested the large and XL configurations of the model.

In Table 3, we report the size of the models in terms of parameters, the type of model, and the size calculated in terms of times the models fit in Llama-2 13B.

TABLE 3. Description of models. The column *Type* refers to the type of model, while the *Comparison* refers to the times the size of the model fits into the Llama-2 13B one.

Model	Parameters	Type	Comparison
Flan T5 - large	780M	Seq2Seq	16.7×
Flan T5 xl	3B	Seq2Seq	4.3×
Falcon	7B	CausalLM	1.9×
Mistral	7B	CausalLM	1.9×
Llama-2	7B	CausalLM	1.9×
Llama-2	13B	CausalLM	1×

C. POSTPROCESSING

Generative models have a generation of maximum new tokens hyperparameter that we set at 64. Setting this parameter, though, does not ensure clean rewrites, and can result in verbose responses that lead to irrelevant responses in the retrieval phase.

To address this issue, we first observe the kind of utterances in the evaluation datasets and decide to eliminate all the generated text that appears after the second full stop or after the first question mark. In fact, in many cases the question comes after a statement (e.g., *Now my garage door opener stopped working. Why?*) or the user request is composed of two statements (e.g. *No, not information about Burger King's acquisition. I want to know how to open a Burger King franchise.*)

In the post-processing phase, we use regular expressions to remove all alien elements relative to the input and the extra generated text.

D. BASELINES

We assess the retrieval effectiveness of original, manually-rewritten, and automatically-rewritten utterances and consider the following rewriting methods and baselines:

- *Original utterances*: raw utterances provided by TREC CAsT.
- *Manual utterances*: manually-rewritten utterances by human annotators provided by TREC CAsT.
- *QuReTeC* [52]: utterances are rewritten with a BiLSTM sequence to sequence model trained for query resolution.
- *CQR self-learn cv* [62]: utterances are generated in two steps, first with a GPT-2 model trained with self-supervised learning to generate contextual utterances containing few information presented in previous utterances. The second step is performed with a GPT-2 model fine-tuned on manual rewrites via five-fold cross-validation.
- *CQR rule-based cv* [62]: utterances are generated in two steps, first with a rule-based approach that deals with omissions and coreference and successively rewritten with a GPT-2 model fine-tuned on manual rewrites via five-fold cross-validation.
- *Prompt E* [30]: although the results by Mao *et al.* are achieved on a different generative model, i.e., GPT-3, we use their prompt in our experimental framework to compare its retrieval performance with ours.
- *ConvGQR* [34]: as they show in their research, they test their rewrites on CAsT 2019 and 2020 using a dense retrieval method, namely ANCE. Given the public unavailability of their rewrites, we replicate their method on our generations and then compare with the results shown on their article.

E. TWO-STAGE RETRIEVAL

To assess and contrast the various utterance rewrites, we index the TREC CAsT collections by eliminating stopwords and implementing Porter's English stemmer. We employ PyTerrier [28] to construct the information retrieval pipeline, which is divided into two phases:

- The initial phase carries out document retrieval on the indexed collection using the DPH weighting model [3], with the raw, manually, and automatically-rewritten utterances;
- The subsequent phase conducts reranking of the top-1000 candidates retrieved by the first phase using the MonoT5 model [38] that is accessible in PyTerrier[6].

We measure the retrieval effectiveness of the first stage and of the second stage using the following metrics: Mean Reciprocal Rank (MRR), Precision@1 (P@1), Normalized Discounted Cumulative Gain@3 (NDCG@3), and Recall@500 (R@500). MRR and NDCG@3 are standard metrics used for evaluation purposes in the TREC CAsT framework, while the others are included to provide a more comprehensive evaluation of the retrieval capabilities of the first-stage (R@500) and the reranking capabilities of the second-stage (P@1).

IV. RESULTS: PROMPTING INSTRUCTION-TUNED LLMS

We now present the experimental results on CAsT 2019 and 2020 datasets achieved by prompting `gpt-3.5-turbo` to evaluate the different rewriting strategies, comparing them with the baselines.

A. FIRST-STAGE RETRIEVAL

Table 4 shows the outcomes of document retrieval using the DPH weighting model [3]. These outcomes relate to the first retrieval stage for both the CAsT 2019 and CAsT 2020 datasets. The performance of our methods and the baselines vary between the results obtained for the original utterances and those that have been manually rewritten.

For CAsT 2019, P5 is the best-performing prompt in terms of MRR while P1 is the best-performing prompt for Preci-

[6]https://github.com/terrierteam/pyterrier_t5

TABLE 4. First-stage retrieval results in terms of MRR, P@1, NDCG@3 and R@500 on CAsT 2019 and CAsT 2020 datasets. In bold, we report the best results achieved for each metric, except Manual. We mark statistically-significant performance gain/loss, calculated with the two-paired *t*-test (*p*-value < 0.05) with Bonferroni correction, of our methods with respect to the QuReTeC and CQR self-learn cv baselines with the symbol † for the first, and ∗ for the latter.

Prompt	CAsT 2019				CAsT 2020			
	MRR	P@1	NDCG@3	R@500	MRR	P@1	NDCG@3	R@500
Manual	0.675∗	0.549	0.400∗	0.737∗	0.622†	0.505†	0.328†	0.668†
Original	0.333∗	0.2254∗	0.162∗	0.382∗	0.218†	0.159†	0.100†	0.253†
P1	0.633	**0.526**	**0.366**	0.645	**0.535**†	**0.423**†	**0.251**	**0.571**
P2	0.589	0.462	0.292	0.578∗	0.484	0.375	0.241	0.549
P3	0.613	0.509	0.336	0.604†	0.458	0.315	0.215	0.501
P4	0.622	0.512	0.345	0.631	0.430	0.332	0.211	0.496
P5	**0.636**	0.515	0.333	0.650	0.478	0.389	0.227	0.513
E	0.584	0.480	0.309	0.577†	0.452	0.356	0.218	0.503
QuReTec	0.625	0.491	0.349	**0.670**	0.440	0.322	0.215	0.516
CQR self-learn cv	0.592	0.468	0.334	0.662	-	-	-	-
CQR rule-based cv	0.563	0.416	0.311	0.657	-	-	-	-

sion@1 and NDCG@3. For R@500. the QuReTec baseline is the best-performing method. When conducting the statistical significance evaluation using a two-paired t-test (p-value <0.05) with the Bonferroni correction [45], the results obtained by our prompts are not statistically different from the state-of-the-art baselines, except for R@500 for P2, P3, and E.

Better results are obtained when rewriting the utterances of the CAsT 2020 evaluation dataset. The best-performing rewriting method is based on P1, where all metrics show significant gains over the QuReTec baseline. For P@1 and MRR, the improvement achieved by P1 is statistically significant compared to the QuReTec baseline, with a 21.6% gain in MRR and 31.7% in P@1. NDCG@3 and R@500 increase by 17.1% and 10.6%, respectively. We remind the reader that P1 also takes into account the generated answers to the previously rewritten questions to produce the current rewriting. In fact, it is important to note that, unlike CAsT 2019 where most relevant concepts could be found in the previous utterances, for CAsT 2020, some missing relevant concepts that complete the context, can be found only in the responses and not in the utterance history. Results show that by generating the answers to the user requests and instructing the model to use them in the rewriting phase, we achieve better results.

B. SECOND-STAGE RETRIEVAL

Table 5 shows the end-to-end results achieved with CAsT 2019 and 2020 when performing document re-ranking using the MonoT5 model in the second-stage retrieval pipeline. We hypothesize that because our rewriting techniques produce verbose and well-formed utterance rewritings, it would be advantageous to use a LLM-based model such as T5, so as to effectively utilize the information added by the `gpt-3.5-turbo` model. We can observe that the performance achieved by the generated rewritings surpasses the results obtained by the CQR and QuReTec competitors for prompts such as P1, P5 for CAsT 2019, and for all prompts for CAsT 2020. The best-performing method for CAsT 2019 is P5, with an MRR of 0.8119 (3.3% increase), P@1 of 0.7283 (5.9% increase), NDCG@3 of 0.5343 that is slightly higher than the one provided by QuRETec, i.e., 0.5330. In line with the first stage, also in the second-stage retrieval, the results are better than the QuReTec baseline, except for R@500. although not statistically significant. When considering the CAsT 2020 evaluation dataset, our rewriting methods show considerable improvements after reranking. In this case, we have a clear winner, i.e., P1, for which all metrics improve over QuReTec in a statistically-significant way. The MRR increases by 25.2%, the P@1 by 31.7%, the NDCG@3 by 27.0%, and the R@500 by 11.5%. Also, for P2, we have a statistically-significant improvement of 22.17% in terms of NDCG@3. Even in the second stage of retrieval, we achieve results as good as—or better than—state-of-the-art competitors, confirming that instructed LLMs are effective in rewriting utterances in a multi-turn conversational setting.

C. ANSWERING OUR RESEARCH QUESTIONS

RQ1. We affirm that using an Instruction-tuned LLM to rewrite utterances helps the effectiveness of the retrieval system. For the CAsT 2020 dataset, we obtain significant improvements over the QuReTeC baseline, while for the CAsT 2019 we achieve the same results, and in some cases, we outperform QuReTeC and the two CQR competitors.

The results achieved also show that, although the LLM was not fine-tuned explicitly for utterance rewriting, it provides competitive results compared to the state-of-the-art. This confirms the ability of these models to perform a variety of tasks via few-shot learning, thus lowering the effort needed for targeting novel tasks. In fact, custom-made models for utterance rewriting in conversational search, i.e., QuReTec, reach worse results on CAsT 2020 than an Instruction-tuned LLM with well-designed prompts. We explain these results as a consequence of the capability of an LLM to deal with different datasets and domains, keeping a rewriting quality higher than other systems trained on limited data, and thus,

TABLE 5. Second-stage retrieval results in terms of MRR, P@1, NDCG@3 and R@500 on CAsT 2019 and CAsT 2020 datasets. In bold, we report the best results achieved for each metric, except Manual. We mark statistically-significant performance gain/loss, calculated with the paired *t*-test (*p*-value < 0.05) with Bonferroni correction, of our corresponding methods with respect to the QuReTeC and CQR self-learn cv baselines with the symbol † for the first, $*$ for the latter.

Prompt	CAsT 2019				CAsT 2020			
	MRR	P@1	NDCG@3	R@500	MRR	P@1	NDCG@3	R@500
Manual	0.885$*$†	0.827$*$†	0.605$*$†	0.772$*$†	0.816†	0.732†	0.538†	0.736†
Original	0.464$*$†	0.399$*$†	0.279$*$†	0.406$*$†	0.3301†	0.221†	0.181†	0.283†
P1	0.791	0.694	0.519	0.697	**0.725**†	**0.639**†	**0.439**†	**0.629**†
P2	0.744	0.636	0.483†	0.635†	0.676	0.596	0.422†	0.609
P3	0.738	0.665	0.487	0.642†	0.602	0.514	0.354	0.560
P4	0.758	0.653	0.516	0.671	0.609	0.524	0.360	0.547
P5	**0.812**	**0.728**	**0.534**	0.706	0.654	0.572	0.405	0.565
E	0.686	0.595	0.451	0.616†	0.616	0.548	0.386	0.557
QuReTec [52]	0.786	0.688	0.533	**0.711**	0.579	0.486	0.345	0.564
CQR self-learn cv [62]	0.778	0.705	0.529	0.694	-	-	-	-
CQR rule-based cv [62]	0.763	0.682	0.511	0.685	-	-	-	-

characterized by a lower generalization power.

RQ2. For what concerns the best way of prompting the LLM, the best results are obtained with P1 for CAsT 2020 and P1 and P5 for CAsT 2019. While for some of the prompts we clearly explicit the scope of the rewriting (e.g. ''[...]for a retrieval system[...]'' in P2), in both P1 and P5 this information is not explicit, suggesting that this kind of instruction is not useful to obtain better rewritings.

Moreover, in both cases, there is a clear indication of how to exploit examples and context from the previous interactions. The difference is that P1 explicitly asks the model to also add previously generated answers to the context and use all the information for generating the rewriting $\hat{u}_i$. This proved particularly effective in the case of CAsT 2020. This could also be the reason why QuReTec underperforms as, by design, it only focuses on the previous utterance and does not integrate the content of the answers for generating the rewriting. Therefore, after establishing the best-performing prompts and observing that they both make use of the context, we can conclude that providing examples can have a significant impact on the model's capabilities in performing the chosen task.

V. RESULTS: FINE-TUNING LLMS

We discuss the results obtained by the different fine-tuned models. We test all the models listed in III-B, and after establishing the best performing one, Llama-2-13B, we test it by lowering the number of bits for the weight representation to 8 and 4, respectively. Before delving into the experimental results, we reemphasize that the primary goal of this research is to establish the quality of fine-tuned models when rewriting conversational queries, and more importantly, to assess whether there is a direct correlation between the model size and the quality of results. In this way, we want to study whether we need such big models or if we could obtain acceptable results while using much smaller models.

A. FIRST-STAGE RETRIEVAL

Per the results discussed in Section IV, we performed two stages of evaluation, the first is done by employing PyTerrier as a retrieval suite, using the DPH weighting model. As a first step, we generate the rewriting for each utterance of CAsT 2019 and 2020.

In Table 6, we report the retrieval results of the rewriting generated with the different models. As for the Instruction-tuned LLMs, we consider MRR, P@1, NDCG@3, and R@500. We observe that the best-performing model for CAsT 2019 is Llama-2-13B, with results close to those established by manually rewritten utterances and not significantly different from them. It should be noted that Llama-2-13B exceeds the results obtained by ChatGPT in the best configurations with an increment of 3.29% for MMR, 2.21% for P@1, 4.37% for NDCG@3 and 9.91% for R@500. Similarly, we observe the results obtained by both Flan T5 configurations. The Flan T5-XL is the second best model, obtaining similar or better results than the one obtained with `gpt-3.5-turbo`, with an increment of 9.9 % for R@500. It is also interesting to note that using an 8-bit quantized version of the same Llama-2-13B model provides good quality results while reducing the model size by four times.

For what concerns CAsT 2020, as observed before in the literature, the retrieval results are lower when compared with the manually rewritten utterances. In this case, we can observe that the results obtained by Prompt 1 with `gpt-3.5-turbo` are the best when considering MRR and P@1, while Llama-2-13B obtains the best results when considering NDCG@3 and R@500 with an increment of 2.1% and 0.8% respectively. In this case, it is worth noting that the results obtained with Llama-2-13B are comparable, if not better than the one obtained with `gpt-3.5-turbo`.

When analyzing the results for both CAsT 2019 and CAsT 2020 for the other models, we can observe that Flan T5-Large obtains remarkable results if taking into account its size (780 millions parameters) as it manages to exceed

gpt-3.5-turbo results for NDCG@3 and R@500. When considering Mistral-7B and Falcon-7B, our results are significantly lower than the chosen comparison. This may be due to the different kinds of data on which the models were originally trained.

B. SECOND-STAGE RETRIEVAL

When analyzing the metrics obtained with the second-stage retrieval reported in Table 7, i.e., a reranking phase performed with T5-Mono, we observe similar results. Regarding CAsT 2019, Llama-2-13B is the best-performing model when considering MMR, P@1, and NDCG@3, obtaining results that are comparable with those obtained with the manually rewritten utterances. Compared with the best ChatGPT prompt results, Llama-2-13B surpassed it for all metrics, with an increase of 7.35% in MRR, 9.67% for P@1, 10.58% for NDCG@3 and 5. 04% for R@500.

When considering R@500, the best-performing model is Flan T5-XL with an increment of 5.1% compared to Prompt 5, the top-performing configuration of gpt-3.5-turbo for this metric.

When considering CAsT 2020, we notice better results than the first-stage retrieval. If we consider P@1, we obtain results that are statistically comparable with those obtained by the Manual baseline for Llama-2-7B and Llama-2-13B, and Prompt 1 with ChatGPT. For MRR, NDCG@3, and R@500, the results are lower than the ones given by the manually rewritten utterances but higher than the QuReTeC baseline and higher than the ones obtained by the best-performing ChatGPT prompt (Prompt 1). Llama-2-13B increases P@1 of 3.03 % and NDCG@3 of 5.22%.

Regarding the first stage, the results obtained by Falcon-7B and Mistral-7B are significantly lower than the baselines for CAsT 2019 and comparable when considering CAsT 2020.

The strong performance of Llama-2 models also on CAsT 2020 suggests that they generalize well across different datasets. Their ability to maintain high precision indicates that the models effectively capture the intent behind user queries, even in challenging datasets where manually rewritten utterances typically perform better. This capability seems to increase with the model size; in fact, the biggest model, i.e., Llama-2-13B, is the one that is best suited to rewrite the queries where it is harder to track back the context.

C. COMPARISON WITH THE STATE-OF-THE-ART

To compare our results with those obtained by ConvGQR [34] (see Subsection III-D) we replicate their evaluation process using ANCE to perform a retrieval phase with the rewrites generated with our models. In Table 8, we report the results obtained.

As we can observe, in line with the results obtained with the other approaches, we have higher results when evaluating CAsT 2019. In this case, we observe that we surpass the results obtained with the manually rewritten utterances. More specifically, Flan T5-XL obtains better results for P@1 (+ 1.81%) and NDCG@3 (+1.82) when the first utterance of each conversation is kept unchanged. Llama-2-13B, on the other hand, manages to obtain better results than the Manual baseline for every metric, i.e. +0.42 % for MRR, +1.81 % for P@1, +2.69 % for NDCG@3 and +0.58 % for R@500. Also, when comparing with ConvGQR, our best model, Llama2-13B, obtains an increase of + 4.96% in terms of MRR and + 9.22% for NDCG@3 for CAsT 2019. When considering CAsT 2020, the same model obtains an increase of 34.60% for MRR and 10.30% for NDGC@3. When observing the results obtained for CAsT 2020, the best-performing models are Llama-13-B on the one hand and gpt-3.5-turbo on the other hand. In line with the results obtained when reranking and with the literature examples, the results are lower than those obtained with the manual rewrite but not statistically different. It is worth noting that for both metrics used to evaluate ConvGQR, MRR, and NDCG@3, we obtain higher results than the ones reported in their study for both CAsT 2019 and 2020. Unfortunately, due to the inaccessibility of their rewrites, we couldn't test the statistical significance with respect to their results.

D. EFFICIENCY STUDIES

We demonstrated that open-source LLMs deliver equivalent or superior performance compared to corporate models such as gpt-3.5-turbo. Among the several advantages provided by replacing ChatGPT with an open-source model — privacy, democratizing scientific research — we can consistently speed up the process of query rewriting. We further delve into efficiency analysis by studying the different execution times entailed by different open-source LLMs. Now we discuss the quality of the rewriting and retrieval performances with respect to the size of the models and the inference time.

Table 9 reports the results of our experimental evaluation. We report the average time to rewrite a query for each model, considering different batch sizes. Recall that the batch size is the number of instances, queries in this case, provided to the model simultaneously. Increasing the batch size is a common approach to speed up both the training and the inference of neural models since a larger batch size allows for increased resource utilization on the GPU, thus reducing the average inference latency. The experiments were performed on an A100 NVIDIA GPU equipped with 80 GB of RAM. Queries from the TREC 2019 collection were used for this experimental step. We perform a warm-up step to avoid any overhead given by GPU initialization. We employ the pytorch library and the torch.cuda.event[7], which allows us to isolate the time spent on inference on the GPU, without taking into account any data transfer costs.

As expected, Table 9 shows the positive impact of larger batch sizes on the average query inference time. Smaller models tend to benefit more from the simultaneous processing of several instances compared to large ones. As an example, increasing the batch size from 1 to 128 ensures a 16× speedup with Flan-t5-Large, while only 8× on Mistral-7b. As the

[7]https://pytorch.org/docs/stable/generated/torch.cuda.Event.html

TABLE 6. First-stage retrieval results in terms of MRR, P@1, NDCG@3 and R@500 on the data sets CAsT 2019 and CAsT 2020. In bold, we report the best results achieved for each metric, except Manual. We mark statistically significant performance gain/loss, calculated with the two-paired *t*-test (*p*-value < 0.05) with Bonferroni correction, of our methods with respect to the Manual and QuReTeC baselines with the symbols † for the first, $*$ for the latter. In the column First, we report whether the first utterance of each conversation was kept unchanged(Orig) or was rewritten by the model(Rewr).

	CAsT 2019				CAsT 2020			
Model	**MRR**	**P@1**	**NDCG@3**	**R@500**	**MRR**	**P@1**	**NDCG@3**	**R@500**
Original	0.333†*	0.225†*	0.162†*	0.382†*	0.218†*	0.159†*	0.100†*	0.253†*
Manual	0.675	0.549	0.400	0.737*	0.622*	0.505*	0.328*	0.668*
Falcon-7B	0.483†*	0.347†*	0.258†*	0.562†*	0.410†	0.312†	0.200†	0.492†
Flan T5-Large	0.609†	0.474	0.345†	0.694†	0.504†	0.404†	0.248†	0.515†
Flan T5-XL	0.633	0.497	0.363	0.708	0.481†	0.380†	0.234†	0.527†
Llama-2-7B	0.602	0.480	0.343	0.660†	0.481†	0.385†	0.225†	0.554†
Llama-2-13B	**0.657**	**0.538**	**0.382**	**0.714**	0.516†	0.409	**0.257**†	**0.575**†
Llama-2-13B 4bit	0.608	0.474	0.344	0.693†	0.506†	0.389†	0.236†	0.543†
Llama-2-13B 8bit	0.628	0.503	0.357	0.697†	0.482†	0.370†	0.237†	0.557†
Mistral-7B	0.358†*	0.243†*	0.150†*	0.378†*	0.365†	0.269†	0.151†*	0.399†*
Prompt 1	0.633	0.526	0.366	0.645†	**0.535***	**0.423**	0.251†	0.571†
Prompt 5	0.636	0.514	0.332†	0.650†	0.477†	0.389	0.227†	0.513†
QuReTeC	0.625	0.491	0.349	0.670†	0.440†	0.322†	0.214†	0.516†
CQR rule based cv	0.563†	0.416†	0.311†	0.657†	-	-	-	-
CQR self learn cv	0.592†	0.468	0.334†	0.662†	-	-	-	-

TABLE 7. Second-stage retrieval results in terms of MRR, P@1, NDCG@3 and R@500 on CAsT 2019 and CAsT 2020 datasets. In bold, we report the best results achieved for each metric, except Manual. We mark statistically significant performance gain/loss, calculated with the two-paired *t*-test (*p*-value < 0.05) with Bonferroni correction, of our methods with respect to the Manual and QuReTeC baselines with the symbols † for the first, $*$ for the latter. In the column First, we report whether the first utterance of each conversation was kept unchanged(Orig) or was rewritten by the model(Rewr).

	CAsT 2019				CAsT 2020			
Model	**MRR**	**P@1**	**NDCG@3**	**R@500**	**MRR**	**P@1**	**NDCG@3**	**R@500**
Original	0.466†*	0.399†*	0.281†*	0.408†*	0.331†*	0.274†*	0.181†*	0.283†*
Manual	0.884*	0.821*	0.599*	0.773*	0.814*	0.721*	0.534*	0.736*
Falcon-7B	0.637†*	0.549†*	0.398†*	0.592†*	0.580†	0.505†	0.337†	0.544†
Flan T5-Large	0.823	0.740	0.552	0.730†	0.683†*	0.596†*	0.416†*	0.583†
Flan T5-XL	0.841	0.757	0.576	**0.742**	0.692†*	0.606†*	0.426†*	0.591†
Llama-2-7B	0.788†	0.711	0.515†	0.698†	**0.728**†*	**0.654***	0.447†*	0.623†
Llama-2-13B	**0.863**	**0.786**	**0.590**	0.742	0.722†*	0.644*	0.458†*	**0.638**†*
Llama-2-13B 4bit	0.815	0.723	0.539	0.726†	0.686†*	0.587†	0.419†*	0.607†
Llama-2-13B 8bit	0.832	0.728	0.554	0.736†	0.689†*	0.596†	0.419†*	0.617†
Mistral-7B	0.575†*	0.480†*	0.344†*	0.440†*	0.539†	0.447†	0.306†	0.471†*
Prompt 1	0.788†	0.688†	0.514†	0.699†	**0.727***	0.635*	0.436†*	0.630†
Prompt 5	0.804†	0.717†	0.534	0.706†	0.651†	0.567†	0.399†	0.566†
QuReTeC	0.795†	0.705†	0.531†	0.713†	0.579†	0.481†	0.338†	0.566†
CQR rule based cv	0.761†	0.676†	0.512†	0.688†	-	-	-	-
CQR self learn cv	0.779†	0.705†	0.535†	0.696†	-	-	-	-

model gets bigger, the performances plateau with a smaller value of the batch size, e.g., Llama-2-7b does not benefit from increasing the batch size from 16 to 32. Moreover, large models can produce an Out-Of-Memory exception if the batch size is too big, given the fact that the combination of the model parameters and the input and activation tensor surpasses the memory capability of the device.

Energy Consumption. Finally, we analyze the energy consumption associated with using larger models, particularly focusing on the energy required for the query re-writing step. In particular, we consider the total energy employed to re-write the 173 queries from the TREC Conversational Assistant Track 2019. We measure GPU energy using the `Zeus` library [60] and present our results in Figure 5. For each model, we calculate total energy consumption while adjusting the batch size. As for execution times (Table 9), we observe significant differences across models. Unsurprisingly, the architectures that are more time-efficient also tend to be more energy-efficient. The plot also shows that energy efficiency improves with increased batch size; since we account for the entire memory consumption to re-write the queries, larger batch sizes help reduce the number of inference runs required. We observe that using Flan-T5-large instead of Llama-2-13B allows saving up to $20\times$ the amount of total energy, highlighting the importance of a careful selection of re-writing system according to the quality requirements.

TABLE 8. Retrieval results in terms of MRR, P@1, NDCG@3 and R@500 on CAsT 2019 and CAsT 2020 datasets using ANCE. In bold, we report the best results achieved for each metric. We mark statistically significant performance gain/loss, calculated with the two-paired *t*-test (*p*-value < 0.05) with Bonferroni correction, of our methods with respect to the Manual and QuReTeC baselines with the symbols † for the first, $*$ for the latter. The underlined results relative to ConvGQR are taken directly from the paper [34]. In the column First, we report whether the first utterance of each conversation was kept unchanged(Orig) or was rewritten by the model(Rewr).

	CAsT 2019				CAsT 2020			
Model	**MRR**	**P@1**	**NDCG@3**	**R@500**	**MRR**	**P@1**	**NDCG@3**	**R@500**
Original	0.420†*	0.364†*	0.247†*	0.228†*	0.290†*	0.240†*	0.150†*	0.189†*
Manual	0.740	0.642	0.462	0.463*	0.724*	0.625*	0.422*	0.534*
Falcon-7B	0.691	0.595	0.424	0.439	0.561†	0.462†	0.323†	0.450†*
Flan T5-Large	0.706	0.613	0.444	0.434	0.583†	0.500†	0.327†	0.447†
Flan T5-XL	0.735	0.653	0.470	0.446	0.586†	0.495†	0.332†	0.449†
Llama-2-7B	0.686	0.590	0.415	0.443	0.584†	0.495†	0.325†	0.472†*
Llama-2-13B	**0.743**	**0.653**	**0.474**	0.466*	0.626†	**0.529**	**0.365***	0.482†*
Llama-2-13B 4bit	0.721	0.624	0.453	0.461	0.619†	0.524	0.333†	0.479†*
Llama-2-13B 8bit	0.714	0.619	0.456	0.458	0.599†	0.490†	0.339†	0.470†*
Mistral-7B	0.541†*	0.445†	0.324†*	0.368†	0.444†	0.346†	0.230†	0.394†
Prompt 1	0.690	0.595	0.430	**0.467***	**0.633**	0.524	0.354	**0.488***
Prompt 5	0.665	0.567	0.431	0.441	0.599†	0.524	0.336†	0.422†
QuReTeC	0.692	0.590	0.429	0.420†	0.528†	0.452†	0.287†	0.386†
CQR rule based cv	0.665	0.567	0.409	0.403†	-	-	-	-

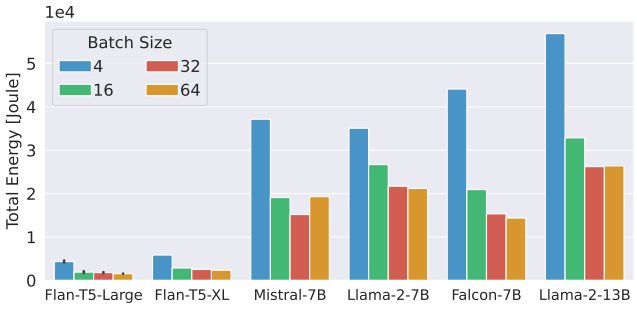


FIGURE 5. Total energy consumed by different models to rewrite the 173 queries from the TREC Conversational Assistant Track 2019, with different batch sizes.

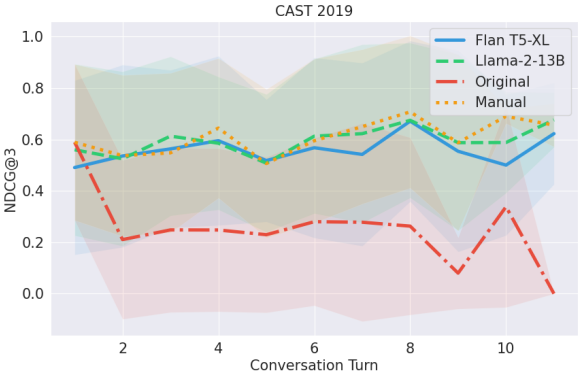


FIGURE 6. Performances of the best-performing models for NDCG@3 after performing the reranking phase for CAsT 2019 throughout the conversational turns. The shaded part represents the standard deviation.

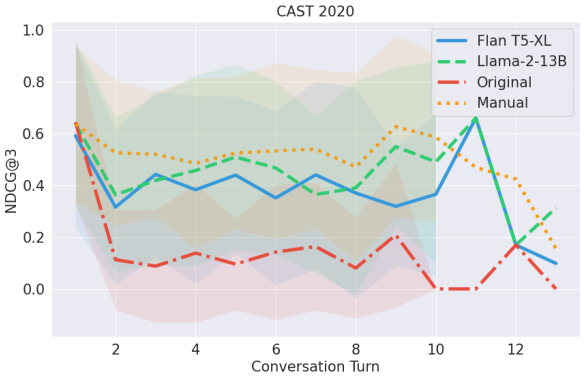


FIGURE 7. Performances of the best-performing models for NDCG@3 after performing the reranking phase for CAsT 2020 throughout the conversational turns. The shaded part represents the standard deviation.

E. ABLATION STUDY

We evaluate the impact of query rewriting in different conversation turns, and we also analyze the average inference time of the models. As we can see in figures 6 and 7, the contribution of the two top models, Flan T5-XL and Llama-2-13B, is quite stable throughout the conversation, showing a peak around the eighth turn for the CAsT 2019 and the seventh for CAsT 2020. For the last one, we can observe the maximum on the 11th turn of the conversation. Still, as we can observe from the image, it is not very indicative since the number of conversations that reach the 11th turn is really low, hence the "unobservable" standard deviation in the figure.

The results shown in the figures also confirm the difficulty encountered when dealing with CAsT 2020 with respect to 2019. In fact, the quality of the generated rewritings decreases with respect to the first turn. This is in line with the nature of the dataset, in which much of the necessary information for a good rewrite is available in the answer passage.

Finally, in Table 9, we show the average inference time for all the models while varying the batch size. As we can observe, both T5-based models require a much shorter inference time than the others. For example, we can see that Flan T5-

XL is, on average, $4.85\times$ faster than Llama-2-7B, while Flan T5-large is $6.24\times$ faster. This shows us that even though not being the best performing models when considering rewrites, both the models based on T5 have a computational cost, both in terms of space and inference time, that is considerably lower than the other models. This suggests that these kinds of models may be further explored and also that it may be possible to achieve even higher results when upscaling the size of such built models.

TABLE 9. Comparison of Average Inference Time at various batch sizes among different large language models.

Batch size Model	1	2	4	8	16	32	64	128
Flan T5-large	284	177	139	83	48	33	22	16
Flan T5-XL	305	240	152	93	61	46	34	33
Llama-2-7B	1524	1072	598	399	272	217	241	-
Mistral-7B	1610	1178	654	421	284	219	217	209
Falcon-7B	2066	1237	716	466	283	203	187	181
Llama-2-13b	2793	1594	952	708	483	429	-	-

F. REWRITING EXAMPLES

We exemplify rewrites produced by the models that are particularly relevant to the current research. Given the fact that the scope of this research is not to generate a well-formed answer to the user query but to study the impact of LLMs on the retrieval phase, we do not study features such as fluency and human readability of the query. Nevertheless, it is important to observe whether the models accurately rewrite some incomplete queries.

For instance, when asked to rewrite the second turn of a conversation about bees (the previous question was "*What are some interesting facts about bees?*"), the models provided interesting results. The raw utterance "*Why doesn't it spoil?*", was referring to the answer provided by the system but none of the models had the answer. Interestingly, three models, i.e., Flan-T5 XL, Llama-2-13B, and Falcon-7B, rewrite the sentence as "*Why doesn't honey spoil?*", making explicit reference to honey, with LLama-2-7B adding also some generic question to the query, i.e., "*Where does honey come from and why doesn't it spoil?*". Among the other models, two got the wrong reference (*hive* and *beewax* respectively), while Mistral-7B inserted part of the answer ("*Why doesn't honey spoil when bees produce a substance that inhibits the growth of harmful microorganisms in the hive and on bee pollen, helping to ensure the bees' survival during times of bacterial attack and disease within the colony."*).

Similarly, when rewriting the second query of a conversation, with the first being "*Which is the biggest commercial plane?*", it is interesting to note how all the models rewrote the second query, i.e., "*What are its operational costs?*". In fact, some of the models (Flan-T5 large, LLama-2-13B 4bit, Mistral and LLama-2-7B) resolved the anaphora by indicating the Boeing 747 as the biggest plane and resulting in rewrites such as "*What are the 747's operational costs?*" with some of them adding more details about the request (i.e., *in terms of fuel, labour and maintenance*). All other models interpret the biggest commercial plane as the Airbus A380, resulting in rewrites such as "*What are the operational costs of the Airbus A380?*". In this case, it is interesting to notice how the parametric memory of the models might be beneficial in rewriting the queries by resolving some references in a more accurate way. The term parametric memory refers to the capability of LLMs to behave as a knowledge base and to store information from training data in their own weights without the need to draw on external data [42].

Overall, while manually observing the generated data, we can affirm that all of the models are able to resolve anaphoras and ellipses proficiently, particularly when the reference is explicit in the previous conversational queries. Furthermore, the fact that, in both cases shown here, the models used internal knowledge to rewrite the query would suggest that some investigation in this sense should be made. In fact, if we manage to extract factual knowledge from the model proficiently, we could further increase the system's capabilities.

G. ANSWERING OUR RESEARCH QUESTIONS

a: RQ3

We confirm that it is possible to increase the quality of rewriting using fine-tuned LLMs compared to the use of Instruction-Tuned LLMs. We show that using fine-tuned models, we obtain better results than the one obtained by ChatGPT with a few-shot approach, particularly when considering LLama-2-13B and Flan T5-XL both for CAsT 2019 and 2020. Moreover, we manage to reduce by many times the size of the rewriting models, achieving better results. At the same time, with our approach, we manage to achieve higher results than those obtained by state-of-the-art techniques such as ConvGQR. This confirms the utility of using a fine-tuning approach to focus the models on a particular task, in our case, conversation query rewriting. This leaves us with the scientific need to further explore such an approach to understand LLMs capabilities better and to achieve even better results while reducing the model size.

b: RQ4

What are the threshold of performance gain and computational costs when using such big models? Our analysis discloses the potential for effective query rewriting using significantly smaller models compared to proprietary LLMs. As shown in the previous sections, our methodology considerably reduces the computational requirements needed for this task, making it feasible on devices beyond high-performance servers. As an example, although our experimental evaluation exclusively utilizes NVIDIA A100 GPUs, models like Flan-T5-Large can easily be employed on consumer-grade GPUs or even on multithreaded CPUs. This allows for personalization of the query rewriting system based on user preferences and historical data without encountering privacy concerns. The custom model can be deployed and updated on the target

edge device, learning from user preferences without posing potential privacy risks.

VI. RELATED WORK

We now overview the prior contributions, partitioning our discussion into conversational search and LLM-focused efforts.

a: Conversational search

The process of query rewriting plays a pivotal role in contemporary web search, as it helps accurately capture user information requirements and boosts retrieval efficiency [17]. This issue is also prevalent in conversational search, where utterances, akin to queries, can often be ambiguous or poorly structured.

The goal of conversational utterance rewriting is to transform a brief request within a conversational context into a comprehensive, context-independent query that addresses anaphoras, ellipses, and other linguistic phenomena [31], [57]. These methods strive to pinpoint terms that were mentioned earlier in the conversation to beneficially expand the current utterance [2], [32], [43], [52]. In this regard, Alianne-jadi *et al.* put forth a unique neural utterance relevance model based on BERT that helps identify utterances pertinent to a given turn [2]. Voskarides et al. [52] treat query rewriting for conversational search as a binary term classification task and present QuReTeC, a Bi-LSTM model that picks out the valuable terms in context to enrich the query.

Other strategies to rewrite utterances leverage a fine-tuned neural model [16], [46], [49], [62], [62] to address coreference and omissions in conversational query rewriting.

In subsequent work, [50] contrast original user questions and human-rewritten questions with questions automatically rewritten by sequence generation models based on GPT-2 and QuReTeC. The authors also demonstrate that simply appending the terms predicted by QuReTeC to the questions rewritten by a sequence-generation model improves the state-of-the-art ranking performance. Mo et al. [34], recently released ConvGQR that combines query rewriting and query expansion, achieving good retrieval results.

Azzopardi et al. [5] propose a conceptual framework for conversational search and recommendation systems, outlining various actions, intents, and critical decision points that arise during conversations. Their objective is to explicitly define these components to facilitate the formalization of research, development, and evaluation of conversational search agents. Fu et al. [13] investigate stopping strategies in conversational search systems, focusing on when users decide to end their interactions. They adapt traditional stopping rules from IR to the sequential, interactive nature of conversations. Meng et al. [33] and Faggioli et al. [11] investigate query performance prediction for conversational search and beyond, addressing limitations in prior work. The first evaluate traditional QPP methods in conversational settings, identifying performance gaps with context-dependent queries and introduce a perplexity-based framework that leverages query rewriting quality, enhancing prediction accuracy. The latter focus on proposing innovative geometric embedding-based QPP metrics for dense conversational search.

Frieder et al. [12] exploit the semantic relatedness of retrieved documents within conversations to create a cache of documents for reducing the latency of responses of the conversational agent. The results show the effectiveness and efficiency of the document embeddings cache in the context of conversational dense retrieval.

In other studies, numerous papers utilize pre-trained language models to represent queries and documents in the same dense latent vector space and then employ the inner product to calculate the relevance score of a document to a given query. In the realm of conversational search, the representation of a query can be computed in two distinct ways. In one scenario, a standalone contextual query understanding module restructures the user query into a rewritten query, leveraging the context history [15], and then a query embedding is computed, for example, using sentence embedding models such as ANCE [56] or STAR [65]. Alternatively, the learned representation function is trained to accept as input the query along with its context history and to generate a query embedding that is more akin to the manual query embeddings [63].

In both scenarios, dense retrieval methods are employed to compute the query-document similarity by deploying efficient nearest neighbor techniques over specialized indexes, like those provided by the FAISS toolkit [20]. Tran et al. [48] created a conversational model using a reinforcement learning approach exploiting the conversational context to generate good quality responses to the user requests. Recent advancements have seen the introduction of models like DPR-CT and ColBERT-QA which enhance retrieval effectiveness by combining dense retrieval techniques with query-aware attention mechanisms [21], [66]. Mao et al. [29] recently introduced ChatRetriever, a new model trained by using contrastive learning to represent conversational sessions in a dense way, showing promising results.

b: Large Language Models

LLMs based on transformer architectures, such as GPT are trained on extensive text data corpora to understand and generate natural language [1], [51]. The pre-trained models produced with unsupervised training [6] can be conveniently fine-tuned for various tasks in a supervised setting. InstructGPT, based on GPT-3, has been fine-tuned using human feedback to enhance its ability to follow user intentions [37]. Bidirectional and Auto-Regressive Transformer (BART) combine the strengths of two established models, that is, BERT and GPT-2, and are trained using a denoising autoencoder approach to comprehend the structure and semantics of the text, as well as generate fluent and coherent text [22].

Another Instruction-tuned LLM model of the GPT family is ChatGPT[8], which is specifically designed for conversational applications [25]. Instruction-tuned LLMs such as ChatGPT are easily adaptable to new tasks and domains, mak-

[8]https://chat.openai.com/

ing them extremely useful in various tasks. Wei *et al.* [55] introduce ChatIE, a framework that uses ChatGPT to perform zero-shot Information Extraction tasks via multi-turn question-answering and assert that their method can achieve remarkable results and outperform some full-shot models across three IE tasks. Recently, Mo et al. [35] introduce ConvSDG, a framework to address the challenge of data scarcity in training conversational dense retrieval systems. Their system generates data at both the dialogue level and query level, supporting unsupervised and semi-supervised settings. Similarly to our scope, Wang et al. [54] provide an in-depth investigation into user response simulation for conversational search, addressing limitations in current user simulators. They show that a smaller finetuned T5 model outperforms existing simulators and large language models (LLMs), such as GPT-4, in generating user-like responses to clarifying questions. Multiple papers, such as [41], [44], [59] offer extensive reviews about the state of the art of Large Language Models, with applications in different areas such as the medical one and a specific focus on the GPT architecture.

According to [47], ChatGPT can achieve comparable or superior results to supervised methods for information retrieval relevance ranking when given domain-specific guidelines. These models, along with other LLMs, have shown remarkable performance in various NLP tasks, and have many applications in different fields, such as medicine, finance, and more. With appropriate instructions, these models can handle a wide range of tasks, making them useful tools for researchers and developers.

Similarly to our first work, [58] used Instruction-tuned models to perform query rewriting and then distillate a T5 for query rewriting. Recent innovations have led to the development of specialized models like Gorilla, which is designed to generate domain-specific responses by incorporating factual knowledge from structured databases, thereby improving the factual accuracy and relevance of the generated content [23].

We use ChatGPT as the Instruction-tuned LLM in our experiments to rewrite the user queries. Furthermore, we exploit finetuning to produce small models to automatically rewrite user requests. We build up on the various available techniques and approaches to create straightforward models that do not require any further development and that manage to reduce the computational costs required to perform such a task. In this flourishing area, our contribution try to establish a sound way to refine Language Models while giving prior importance to usability while maintaining good output quality.

VII. DISCUSSION

Our results demonstrate that fine-tuned LLMs, particularly Llama-2-13B, can achieve retrieval performance comparable to or better than ChatGPT (`gpt-3.5-turbo`) on the TREC CAsT 2019 and 2020 datasets. Specifically, Llama-2-13B outperforms ChatGPT in first-stage retrieval on all evaluation metrics for CAsT 2019 and achieves comparable results for CAsT 2020. In particular, the quantized versions of Llama-2-13B (8-bit and 4-bit) and smaller models of the Flan family also deliver competitive performance while significantly reducing model size.

In particular, compared to state-of-the-art approaches such as ConvGQR [36], our fine-tuned models achieve superior performance on both the CAsT 2019 and the CAsT 2020 datasets. During their experiments, they also used a version of T5 [39] to perform query rewriting and query expansion. Similarly, QuReTeC and CQR used two models, bert-base-uncased and GPT-2, to enhance the user's queries. While the first expands the queries by selecting appropriate terms, the second rewrites the whole query with a model that is approximately twice the size of Flan-T5 large (1.5B parameters vs 780M). This suggests that a good fine-tuning process can notably impact the model's performance. Overall, our approach, while straightforward and conceptually simpler, manages to improve the performances with respect to the previous work taken under consideration. Moreover, we firmly believe that the performance could be further improved by testing new ways of fine-tuning and model selection during the rewriting phase.

Our findings align with previous research emphasizing the potential of fine-tuning smaller models for specific tasks. Liu et al. [26] demonstrated that task-specific fine-tuning substantially improves model performance in natural language understanding tasks. Similarly, [27] showed that fine-tuned transformer models outperform larger, more general models in information retrieval contexts. The success of Flan T5 models, particularly Flan T5-XL, further supports the idea that smaller models can be effectively fine-tuned for query rewriting tasks. Despite being approximately $4.3\times$ smaller than Llama-2 13B, Flan T5-XL achieved retrieval results comparable to or better than those of ChatGPT. This observation is consistent with the findings of [40], who highlighted the versatility of the T5 model across various natural language processing tasks when fine-tuned. Our efficiency studies revealed that smaller models like Flan T5-Large and Flan T5-XL not only reduce computational resources but also significantly decrease inference times. This is crucial for practical applications where computational efficiency and scalability are important considerations. This suggests that fine-tuning LLMs on conversational query rewriting datasets like QReCC can be more effective than relying on generic instruction-tuned models or traditional query reformulation techniques.

VIII. CONCLUSIONS

This study addresses the critical task of Conversational Query Rewriting in the context of AI development. We investigated how to rewrite conversational search queries by taking advantage of the generative power of Instruction-Tuned LLMs. To do so, we selected `gpt-3.5-turbo`, and devised the most proficient way to prompt it to obtain valuable rewritings by testing zero, one, and few-shots approaches. To evaluate such rewrites, we selected the datasets CAsT 2019 and 2020 that were specifically built to evaluate conversational systems. We showed that rewrites produced in such a way manage to

overcome results obtained by state-of-the-art tools such as QuReTeC and CQR.

After establishing the rewriting capabilities of ChatGPT, we wanted to understand whether it may be beneficial to fine-tune LLMs specifically for the Conversational Query Rewriting task. To do so, we selected a few open-source LLMs such as LLama-2 7B and 13B, Falcon-7B, Mistral-7B, and Flan T5 in the large and XL versions. We selected the QReCC dataset as the training set for the fine-tuning phase. After performing a fine-tuning phase for each of the models, we produced rewrites for the CAsT 2019 and 2020 datasets. The first achievement of this research was to establish that such models, specifically fine-tuned for conversational query rewriting, can overcome the results obtained by `gpt-3.5-turbo`.

The last goal of this effort was to understand which models are the most proficient when addressing this task and especially whether and how the retrieval phase is impacted by the size of the models. In this sense, we observed that the best-performing model is LLama-2 in its 13B configuration, which overcomes the results obtained with ChatGPT. When replicating the experiments of a state-of-the-art approach such as ConvGQR, we managed to obtain better results for both datasets. However, the most interesting results we obtained are relative to both the Flan T5 models and the quantized versions of Llama-2-13B. In fact, even if they obtain lower results than the ones achieved by LLama-2 13B, they still achieve remarkable results with a significant gain in the computational power required to perform rewriting. In particular, Flan T5, large and XL, require a much shorter time when running the inference phase and a smaller amount of memory to be loaded. Unlike Flan T5 models, the quantized Llama-2-13B versions are only loaded with a 4-8 bit configuration, meaning that the training phase required the resources to train the full version of Llama-2-13B.

A. IMPLICATIONS

The ability to use smaller, fine-tuned models for conversational query rewriting has significant implications for the field of information retrieval. It enables the deployment of efficient and scalable retrieval systems without compromising performance. This is particularly important in real-world applications where computational resources may be limited.

Moreover, our approach supports the democratization of AI technologies by making high-performing models more accessible. Organizations and researchers with limited resources can leverage fine-tuned, smaller models to achieve state-of-the-art performance without the need for extensive computational infrastructure.

B. LIMITATIONS AND FUTURE WORK

While our study presents promising results, there are limitations that should be addressed in future work. One limitation is the reliance on specific datasets like QReCC for fine-tuning. Expanding the fine-tuning process to include a more diverse range of conversational datasets could further enhance model generalizability. In this respect, it could be useful to automatically generate data using bigger LLMs. In this way, we could distill the rewriting capabilities and knowledge contained in the LLM and pass it through a smaller one. Additionally, although quantized models show competitive performance, there remains a performance gap compared to their full-precision counterparts. Future research could explore advanced quantization techniques or other model compression methods to bridge this gap. Understanding the trade-offs between model size, performance, and computational efficiency is also an important area for further investigation. Exploring early exit strategies and other efficiency optimization techniques could lead to even more practical and efficient conversational query rewriting systems. Finally, we plan on studying the impact of the model over the utterance in order to adapt the model to use with respect to the query. In this way, we want to further reduce energy consumption while further increasing the performance.

REFERENCES

[1] R. Alec, N. Karthik, S. Tim, and S. Ilya. Improving language understanding by generative pre-training. 2018.

[2] M. Aliannejadi, M. Chakraborty, E. A. Ríssola, and F. Crestani. Harnessing evolution of multi-turn conversations for effective answer retrieval. In *Proceedings of the 2020 Conference on Human Information Interaction and Retrieval*, CHIIR '20, pages 33–42. Association for Computing Machinery, 2020.

[3] G. Amati and C. J. Van Rijsbergen. Probabilistic models of information retrieval based on measuring the divergence from randomness. *ACM Trans. Inf. Syst.*, 20(4):357–389, oct 2002.

[4] R. Anantha, S. Vakulenko, Z. Tu, S. Longpre, S. Pulman, and S. Chappidi. Open-domain question answering goes conversational via question rewriting. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

[5] L. Azzopardi, M. Dubiel, M. Halvey, and J. Dalton. A conceptual framework for conversational search and recommendation: Conceptualizing agent-human interactions during the conversational search process. *arXiv preprint*, cs.IR, 2024.

[6] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020.

[7] J. Dalton, C. Xiong, and J. Callan. CAsT 2020: The conversational assistance track overview. TREC'20, Virtual, 2020.

[8] J. Dalton, C. Xiong, and J. Callan. TREC CAsT 2021: The conversational assistance track overview. TREC'21, Virtual, 2021.

[9] J. Dalton, C. Xiong, V. Kumar, and J. Callan. CAsT-19: A dataset for conversational information seeking. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, July 2020.

[10] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

[11] G. Faggioli, N. Ferro, C. I. Muntean, R. Perego, and N. Tonellotto. A geometric framework for query performance prediction in conversational search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 1355–1365, New York, NY, USA, 2023. Association for Computing Machinery.

[12] O. Frieder, I. Mele, C. I. Muntean, F. M. Nardini, R. Perego, and N. Tonellotto. Caching historical embeddings in conversational search. *ACM Trans. Web*, 18(4), Oct. 2024.

[13] X. Fu, M. Perez-Ortiz, and A. Lipani. An analysis of stopping strategies in conversational search systems. In *Proceedings of the 2024 ACM SIGIR International Conference on the Theory of Information Retrieval (ICTIR '24)*, pages 1–11. ACM, 2024.

[14] E. Galimzhanova, C.-I. Muntean, F. M. Nardini, R. Perego, and G. Rocchietti. Rewriting conversational utterances with instructed large language models. In *IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology*, 2023.

[15] J. Gao, C. Xiong, P. Bennett, and N. Craswell. Neural approaches to conversational information retrieval. *CoRR*, abs/2201.05176, 2022.

[16] J. Hao, Y. Liu, X. Fan, S. Gupta, S. Soltan, R. CHADA, P. Natarajan, E. Guo, and G. Tur. Cgf: Constrained generation framework for query rewriting in conversational ai. In *EMNLP 2022*, 2022.

[17] Y. He, J. Tang, H. Ouyang, C. Kang, D. Yin, and Y. Chang. Learning to rewrite queries. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1443–1452, 2016.

[18] F. Jiang, C. Qin, K. Yao, C. Fang, F. Zhuang, H. Zhu, and H. Xiong. Enhancing question answering for enterprise knowledge bases using large language models. *ArXiv*, abs/2404.08695, 2024.

[19] H. Jin, Y. Zhang, D. Meng, J. Wang, and J. Tan. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *CoRR*, abs/2403.02901, 2024.

[20] J. Johnson, M. Douze, and H. Jegou. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(03):535–547, 2021.

[21] O. Khattab and H. Zamani. Colbert-qa: Combining dense and sparse representations for question answering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 30–39, 2022.

[22] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, 2019.

[23] H. Li, J. Zhang, Y. Wang, and L. Chen. Gorilla: Large language models with knowledge integration. *arXiv preprint arXiv:2305.12286*, 2023.

[24] X. Li, Y. Zhou, and Z. Dou. Unigen: A unified generative framework for retrieval and question answering with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(8):8688–8696, Mar. 2024.

[25] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu, Z. Wu, D. Zhu, X. Li, N. Qiang, D. Shen, T. Liu, and B. Ge. Summary of chatgpt/gpt-4 research and perspective towards the future of large language models, 2023.

[26] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[27] S. MacAvaney, A. Yates, A. Cohan, and N. Goharian. CEDR: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1104, 2019.

[28] C. Macdonald, N. Tonellotto, S. MacAvaney, and I. Ounis. PyTerrier: Declarative experimentation in python from BM25 to dense retrieval. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pages 4526–4533. Association for Computing Machinery, 2021.

[29] K. Mao, C. Deng, H. Chen, F. Mo, Z. Liu, T. Sakai, and Z. Dou. Chatretriever: Adapting large language models for generalized and robust conversational dense retrieval. *ArXiv*, abs/2404.13556, 2024.

[30] K. Mao, Z. Dou, H. Chen, F. Mo, and H. Qian. Large language models know your contextual search intent: A prompting framework for conversational search, 2023.

[31] I. Mele, C. I. Muntean, F. M. Nardini, R. Perego, N. Tonellotto, and O. Frieder. Topic propagation in conversational search. In J. X. Huang, Y. Chang, X. Cheng, J. Kamps, V. Murdock, J. Wen, and Y. Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 2057–2060. ACM, 2020.

[32] I. Mele, C. I. Muntean, F. M. Nardini, R. Perego, N. Tonellotto, and O. Frieder. Adaptive utterance rewriting for conversational search. *Inf. Process. Manag.*, 58(6):102682, 2021.

[33] C. Meng. Query performance prediction for conversational search and beyond. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24)*, page 3077. ACM, 2024.

[34] F. Mo, K. Mao, Y. Zhu, Y. Wu, K. Huang, and J.-Y. Nie. ConvGQR: Generative query reformulation for conversational search. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4998–5012, Toronto, Canada, July 2023. Association for Computational Linguistics.

[35] F. Mo, B. Yi, K. Mao, C. Qu, K. Huang, and J.-Y. Nie. Convsdg: Session data generation for conversational search. In *Companion Proceedings of the ACM Web Conference 2024 (WWW '24 Companion)*, pages 1634–1642. ACM, 2024.

[36] J. Mo, Y. Liu, and M. Zhang. ConvGQR: Conversational query reformulation with reinforcement learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 123–134, 2023.

[37] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022.

[38] R. Pradeep, R. Nogueira, and J. J. Lin. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *ArXiv*, abs/2101.05667, 2021.

[39] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), jan 2020.

[40] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

[41] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, J. Ahmad, M. E. Ali, and S. Azam. A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access*, 12:26839–26874, 2024. Conference Name: IEEE Access.

[42] A. Roberts, C. Raffel, and N. Shazeer. How much knowledge can you pack into the parameters of a language model? In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online, Nov. 2020. Association for Computational Linguistics.

[43] G. Rocchietti, O. Frieder, C. I. Muntean, F. M. Nardini, and R. Perego. Commonsense injection in conversational systems: An adaptable framework for query expansion. In *IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology*, 2023.

[44] S. Sai, A. Gaur, R. Sai, V. Chamola, M. Guizani, and J. J. P. C. Rodrigues. Generative AI for Transformative Healthcare: A Comprehensive Study of Emerging Models, Applications, Case Studies, and Limitations. *IEEE Access*, 12:31078–31106, 2024. Conference Name: IEEE Access.

[45] P. Sedgwick. Multiple significance tests: the bonferroni correction. *BMJ (online)*, 344:e509–e509, 01 2012.

[46] H. Su, X. Shen, R. Zhang, F. Sun, P. Hu, C. Niu, and J. Zhou. Improving multi-turn dialogue modelling with utterance ReWriter. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 22–31. Association for Computational Linguistics, 2019.

[47] W. Sun, L. Yan, X. Ma, P. Ren, D. Yin, and Z. Ren. Is chatgpt good at search? investigating large language models as re-ranking agent, 2023.

[48] Q.-D. L. Tran, A.-C. Le, and V.-N. Huynh. Enhancing Conversational Model With Deep Reinforcement Learning and Adversarial Learning. *IEEE Access*, 11:75955–75970, 2023. Conference Name: IEEE Access.

[49] S. Vakulenko, S. Longpre, Z. Tu, and R. Anantha. Question rewriting for conversational question answering. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 355–363. ACM, 2021.

[50] S. Vakulenko, N. Voskarides, Z. Tu, and S. Longpre. A comparison of question rewriting methods for conversational passage retrieval. In D. Hiemstra, M.-F. Moens, J. Mothe, R. Perego, M. Potthast, and F. Sebastiani, editors, *Advances in Information Retrieval*, Lecture Notes in Computer Science, pages 418–424. Springer International Publishing, 2021.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *arXiv*, 2017.

[52] N. Voskarides, D. Li, P. Ren, E. Kanoulas, and M. de Rijke. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 921–930, New York, NY, USA, 2020. Association for Computing Machinery.

[53] L. Wang, C. Lyu, T. Ji, Z. Zhang, D. Yu, S. Shi, and Z. Tu. Document-level machine translation with large language models. In H. Bouamor, J. Pino, and K. Bali, editors, *Proceedings of the 2023 Conference on Empirical*

Methods in Natural Language Processing, pages 16646–16661, Singapore, Dec. 2023. Association for Computational Linguistics.

[54] Z. Wang, Z. Xu, V. Srikumar, and Q. Ai. An in-depth investigation of user response simulation for conversational search. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*, pages 1407–1418. ACM, 2024.

[55] X. Wei, X. Cui, N. Cheng, X. Wang, X. Zhang, S. Huang, P. Xie, J. Xu, Y. Chen, M. Zhang, Y. Jiang, and W. Han. Zero-shot information extraction via chatting with chatgpt, 2023.

[56] L. Xiong, C. Xiong, Y. Li, K. Tang, J. Liu, P. N. Bennett, J. Ahmed, and A. Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *CoRR*, abs/2007.00808, 2020.

[57] L. Yang, H. Zamani, Y. Zhang, J. Guo, and W. B. Croft. Neural matching models for question retrieval and next question prediction in conversation. ArXiv Preprint 1707.05409, 2017.

[58] F. Ye, M. Fang, S. Li, and E. Yilmaz. Enhancing conversational search: Large language model-aided informative query rewriting. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5985–6006, Singapore, Dec. 2023. Association for Computational Linguistics.

[59] G. Yenduri, M. Ramalingam, G. C. Selvi, Y. Supriya, G. Srivastava, P. K. R. Maddikunta, G. D. Raj, R. H. Jhaveri, B. Prabadevi, W. Wang, A. V. Vasilakos, and T. R. Gadekallu. GPT (Generative Pre-Trained Transformer)— A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions. *IEEE Access*, 12:54608–54649, 2024. Conference Name: IEEE Access.

[60] J. You, J.-W. Chung, and M. Chowdhury. Zeus: Understanding and optimizing {GPU} energy consumption of {DNN} training. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 119–139, 2023.

[61] S. Yu, J. Liu, J. Yang, C. Xiong, P. Bennett, J. Gao, and Z. Liu. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 1933–1936. Association for Computing Machinery, 2020.

[62] S. Yu, J. Liu, J. Yang, C. Xiong, P. Bennett, J. Gao, and Z. Liu. Few-shot generative conversational query rewriting. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1933–1936, New York, NY, USA, 2020. Association for Computing Machinery.

[63] S. Yu, Z. Liu, C. Xiong, T. Feng, and Z. Liu. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 829–838, New York, NY, USA, 2021. Association for Computing Machinery.

[64] H. Zamani, J. R. Trippas, J. Dalton, and F. Radlinski. Conversational information seeking. *Foundations and Trends® in Information Retrieval*, 17(3-4):244–456, 2023.

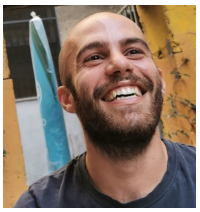
[65] J. Zhan, J. Mao, Y. Liu, J. Guo, M. Zhang, and S. Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1503–1512, New York, NY, USA, 2021. Association for Computing Machinery.

[66] J. Zhan, J. Mao, Y. Liu, and S. Ma. Dense passage retrieval with contextual term expansion. *arXiv preprint arXiv:2204.04979*, 2022.

[67] B. Zhang, B. Haddow, and A. Birch. Prompting large language model for machine translation: A case study. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR, 23–29 Jul 2023.

[68] B. Zhang, H. Yang, T. Zhou, M. Ali Babar, and X.-Y. Liu. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, ICAIF '23, page 349–356, New York, NY, USA, 2023. Association for Computing Machinery.

[69] W. Zhang, Y. Deng, B. Liu, S. Pan, and L. Bing. Sentiment analysis in the era of large language models: A reality check. In K. Duh, H. Gomez, and S. Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3881–3906, Mexico City, Mexico, June 2024. Association for Computational Linguistics.



GUIDO ROCCHIETTI is a PhD Student at University of Pisa and ISTI-CNR in Pisa, Italy. He earned his bachelor's degree in language and literature from the University of Turin in 2017. In 2021 he obtained his master's degree in Digital Humanities at the University of Pisa. In 2021, he then started a PhD program as a student of the Italian National Ph.D. in Artificial Intelligence for Society at the National Research Council (CNR) of Pisa at the HPCLab in the ISTI department. His research fields include Natural Language Processing, particularly Conversational Systems and Summarization. He is now collaborating on two European Union Projects. He is the author of multiple publications in the field. Mr. Rocchietti is currently contracted as a fellow researcher at ISTI-CNR in Pisa.

COSIMO RULLI is a researcher with ISTI-CNR in Pisa, Italy. His research interests include deep learning, model compression, and efficiency in information retrieval. He got his Ph.D. in 2023 with a thesis on Deep Neural Network Compression. He is a co-recipient of the ACM SIGIR 2024 Best Paper Runner-up Award. He is a reveiwer at ACM TOIS, IEEE, PMC, and he is a committee member of SIGIR, ECIR, CIKM, and WSDM.

FRANCO MARIA NARDINI is a Senior Researcher with ISTI-CNR in Pisa, Italy. His research interests are focused on Web Information Retrieval, Machine Learning, and Data Mining. He authored over 100 papers in peer-reviewed international journals, conferences, and other venues. In the past, he has been Program Committee Co-Chair of SPIRE 2023, Tutorial Co-Chair of ACM WSDM 2021, Demo Papers Co-Chair of ECIR 2021. He is a co-recipient of the ACM SIGIR 2024 Best Paper Runner-up Award, the ECIR 2022 Industry Impact Award, the ACM SIGIR 2015 Best Paper Award, and the ECIR 2014 Best Demo Paper Award. He is a member of the editorial board of ACM TOIS and a program committee member of SIGIR, ECIR, SIGKDD, CIKM, WSDM, IJCAI, and ECML-PKDD.

CRISTINA IOANA MUNTEAN is a researcher at ISTI-CNR, Pisa (Italy). Her main research interests are in Information Retrieval and Machine Learning with applications to web search and information retrieval in general. She is particularly interested in conversational search using neural and classic IR models, dense retrieval, and the efficient exploitation of large language models. She co-chaired MICROS, a mixed-initiative conversational search workshop at ECIR and CIKM.She authored more than 40 papers, winning the best paper honorable mention at SIGIR 2023, and has 2000 citations on Google Scholar. She is an active member in the SIGIR, ECIR, CIKM, WSDM, and The Web Conference communities as a paper author and as part of the program committees.

RAFFAELE PEREGO is a research director at ISTI-CNR. His main research interests are in efficient algorithms and machine learning techniques for managing, analyzing, and searching large amounts of data. He published more than 200 papers on these topics in peer-reviewed journals and proceedings of international conferences. He chaired ACM SIGIR in 2016 and ECIR in 2021. He serves in the senior program committees of the top-tier conferences in his research area (ACM SIGIR, ACM CIKM, ACM WSDM, ECIR, WWW) and in the Ph.D. board of the Italian National Ph.D. program on Artificial Intelligence.

OPHIR FRIEDER is an Inductee of the Florida Inventors Hall of Fame. a Fellow of the AAAS, ACM, AIMBE, IEEE, and NAI, an Inaugural Member of the ACM SIGIR Academy, and a Member of both Academia Europaea and the European Academy of Sciences and Arts. His research focuses on scalable information processing systems with particular emphasis on health informatics. He is a member of the computer science faculty at Georgetown University and the biostatistics, bioinformatics and biomathematics faculty in the Georgetown University Medical Center.